Claims

What is claimed is:

1. A method comprising:

in a processor based system where a plurality of processors share processor execution resources, in response to a first processor in the plurality of processors being scheduled to enter an idle state, making a processor execution resource previously reserved for the first processor available to a second processor in the plurality of processors.

2. The method of claim 1 further comprising reserving the processor execution resource for the first processor in response to the first processor being scheduled to execute a task.

3. The method of claim 2 wherein each of the plurality of processors is a logical processor of the processor based system.

4. The method of claim 3 wherein the first processor being scheduled to enter an idle state further comprises the first processor executing a processor instruction requesting the first processor to enter an idle state.

5. The method of claim 4 wherein making the processor execution resource previously reserved for the first processor available to a second processor further comprises releasing the processor execution resource into a common pool of processor execution resources accessible from the second processor.

6. The method of claim 5 wherein the first processor being scheduled to execute a task further comprises the first processor receiving a wake up signal.

7. The method of claim 6 wherein the processor execution resource previously reserved for the first processor further comprises the processor execution resource previously statically allocated to the first processor; and wherein releasing the processor execution resource into a common pool of processor execution resources further comprises de-allocating the processor execution resource.

8. The method of claim 6 wherein the processor execution resource previously reserved for the first processor further comprises the processor execution resource previously locked by the first processor; and wherein releasing the processor execution resource into a common pool of processor execution resources further comprises the first processor unlocking the processor execution resource.

9. The method of claim 6 wherein the common pool of processor execution resources comprises a translation lookaside buffer and the processor execution resource is a translation cache entry from the translation lookaside buffer.

10. A processor comprising:

a plurality of logical processors; and

an instruction set, the instruction set comprising one or more instructions which when executed by a first logical processor, cause the first logical processor to make a processor execution resource previously reserved for the first processor available to a second processor in the plurality of processors in response to the first logical processor being scheduled to enter an idle state.

11. The processor of claim 10 wherein to the first logical processor being scheduled to enter an idle state further comprises the first processor executing a processor instruction requesting the first logical processor to enter an idle state.

12. The processor of claim 11 wherein causing the first logical processor to make the processor execution resource previously reserved for the first logical processor available to a second logical processor further comprises releasing the processor execution resource into a common pool of processor execution resources accessible from the second logical processor.

13. The processor of claim 12 wherein the processor execution resource previously reserved for the first logical processor further comprises the processor execution resource previously statically allocated to the first logical processor; and wherein releasing the processor execution resource into a common pool of processor execution resources further comprises de-allocating the processor execution resource.

14. The processor of claim 12 wherein the processor execution resource previously reserved for the first logical processor further comprises the processor execution resource previously statically allocated to the first logical processor; and wherein releasing the processor execution resource into a common pool of processor execution resources further comprises the first processor unlocking the processor execution resource.

15. A system comprising:

a processor, the processor comprising

a plurality of logical processors; and

an instruction set, the instruction set comprising one or more instructions which when executed by a first logical processor, cause the first logical processor to make a processor execution resource previously reserved for the first processor available to a second processor in the plurality of processors in response to the first logical processor being scheduled to enter an idle state;

firmware to schedule the first logical processor to enter an idle state; and

a bus to interconnect the firmware and the processor.

16. The system of claim 15 wherein the first logical processor being scheduled to enter an idle state further comprises the first processor executing a processor instruction requesting the first logical processor to enter an idle state.

17. The system of claim 16 wherein causing the first logical processor to make the processor execution resource previously reserved for the first logical processor available to a second logical processor further comprises releasing the processor execution resource into a common pool of processor execution resources accessible from the second logical processor.

18. The system of claim 17 wherein the processor execution resource previously reserved for the first logical processor further comprises the processor execution resource previously statically allocated to the first logical processor; and wherein releasing the processor execution resource into a common pool of processor execution resources further comprises de-allocating the processor execution resource

19. The system of claim 17 wherein the processor execution resource previously reserved for the first logical processor further comprises the processor execution resource previously statically allocated to the first logical processor; and wherein releasing the processor execution resource into a common pool of processor execution resources further comprises the first processor unlocking the processor execution resource.

20. A machine accessible medium having stored thereon data which when accessed by a machine causes the machine to perform a method, the method comprising:
in a processor based system where a plurality of processors share processor execution resources, in response to a first processor in the plurality of processors being scheduled to enter an idle state, making a processor execution resource previously reserved for the first processor available to a second processor in the plurality of processors.

21. The machine accessible medium of claim 20 further comprising reserving the processor execution resource for the first processor in response to the first processor being scheduled to execute a task.

22. The machine accessible medium of claim 21 wherein each of the plurality of processors is a logical processor of the processor based system.

23. The machine accessible medium of claim 22 wherein the first processor being scheduled to enter an idle state further comprises the first processor executing a processor instruction requesting the first processor to enter an idle state.

24. The machine accessible medium of claim 23 wherein making the processor execution resource previously reserved for the first processor available to a second processor further comprises releasing the processor execution resource into a common pool of processor execution resources accessible from the second processor.

25. The machine accessible medium of claim 24 wherein the first processor being scheduled to execute a task further comprises the first processor receiving a wake up signal.

26. The machine accessible medium of claim 25 wherein the processor execution resource previously reserved for the first processor further comprises the processor execution resource previously statically allocated to the first processor; and wherein releasing the processor execution resource into a common pool of processor execution resources further comprises de-allocating the processor execution resource.

27. The machine accessible medium of claim 25 wherein the processor execution resource previously reserved for the first processor further comprises the processor execution resource previously locked by the first processor; and wherein releasing the processor execution resource into a common pool of processor execution resources further comprises the first processor unlocking the processor execution resource.

28. The machine accessible medium of claim 25 wherein the common pool of processor execution resources comprises a translation lookaside buffer and the processor execution resource is a translation cache entry from the translation lookaside buffer.